A Dynamical Systems Approach to Energy Disaggregation

Roy Dong, Lillian Ratliff, Henrik Ohlsson, and S. Shankar Sastry

Abstract— Energy disaggregation, also known as non-intrusive load monitoring (NILM), is the task of separating aggregate energy data for a whole building into the energy data for individual appliances. Studies have shown that simply providing disaggregated data to the consumer improves energy consumption behavior. However, placing individual sensors on every device in a home is not presently a practical solution. Disaggregation provides a feasible method for providing energy usage behavior data to the consumer which utilizes currently existing infrastructure. In this paper, we present a novel framework to perform the energy disaggregation task. We model each individual device as a single-input, single-output system, where the output is the power consumed by the device and the input is the device usage. In this framework, the task of disaggregation translates into finding inputs for each device that generates our observed power consumption. We describe an implementation of this framework, and show its results on simulated data as well as data from a small-scale experiment.

I. INTRODUCTION

This paper is motivated by the need of efficient energy management solutions for the retail distribution domain of smart grid. Here, under the term retail distribution domain, we mean the interactions between the local distributors, e.g. the utility companies, and customers, e.g. building occupants. Usage of smart energy management devices has enabled new functionalities and has brought the potential for increased energy efficiency via real-time control and monitoring.

Currently, we focus on commercial and residential buildings. Commercial and residential buildings are major users of energy in the developed world. Buildings account for 20-40% of total energy consumption [1]. We seek to provide customers with individual device power consumption information. Studies have shown that simply providing such data improves the consumer's energy consumption behavior [2].

Current monitoring methods measure total consumption for a building. Placing individual sensors on every device in a home is not presently a practical solution. Disaggregation, also known as non-intrusive load monitoring (NILM), is the task of separating aggregate energy data for a whole

R. Dong, L. Ratliff, H. Ohlsson, and S. Sastry are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA, {roydong, ratliffl, ohlsson, sastry}@eecs.berkeley.edu.

H. Ohlsson is also with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden.

building into the component energy data for individual devices, e.g. refrigerators, stovetops, washing machines, &c. Disaggregation provides a feasible method for providing energy usage behavior data to the consumer, thereby allowing them to identify behavioral trends or device malfunctions that lead to inefficiencies, without requiring major infrastructural changes such as the addition of individual sensors on each device or power receptacle.

Outside of informing consumers about ways to improve energy efficiency, disaggregation presents an opportunity for utility companies to strategically market products to consumers. It is now common practice for companies to monitor our online activity and then present advertisements which are targeted to our interests. This is known as 'personalized advertising'. Disaggregation of energy data provides a means to similarly market products to consumers. This leads to the question of user privacy and the question of ownership with regards to power consumption information. Treatment of the issue of consumer privacy in the smart grid is outside the scope of this paper. However, this is discussed in [3].

Additionally, disaggregation also presents opportunities for improved control. Many devices, such as heating, ventilation, and air conditioning (HVAC) units in residential and commercial buildings implement control policies that are dependent on real-time measurements. Disaggregation can provide information to controllers about system faults, such as device malfunction, which may result in inefficient control. It can also provide information about energy usage which is informative for demand response programs.

We focus on designing disaggregation methods by using dynamical models of devices and formulating the disaggregation problem in an optimal control framework. By working within the dynamical systems and optimal control framework, we hope that our algorithms will lend themselves to easy integration into current real-time optimal control of smart devices within the buildings and for facilitating the implementation of flexible demand response mechanisms by utilities. We designed and set up an experiment to collect data which we use for disaggregation.

The rest of the paper proceeds as follows. In Section II, we discuss the relevant background and existing literature. In Section III, we describe our dynamical system framework for disaggregation, and implementation methods. In Section IV, we test our implementation on simulated data and show results. In Section V, we describe the experimental setup for collecting energy data and discuss the results of the proposed disaggregation method on data from a small-scale experiment. In Section VI, we make concluding remarks and discuss future work.

II. BACKGROUND

The problem of non-intrusive load monitoring and the existing hardware for non-intrusive load monitoring has been studied extensively in the literature (see [4], [5]). The general consensus is that non-intrusive load monitoring is a method to present the consumer with information that makes them aware of their usage and potentially provides them insight into how to improve the efficiency of their usage. Further, the technology to perform non-intrusive load monitoring is becoming widely available. Hence, there is a need for flexible and efficient disaggregation algorithms.

Disaggregation of energy data has emerged as one possible solution for identifying consumer behavior patterns and device malfunctions which lead to inefficient usage of energy. The goal of the current disaggregation literature is to present methods for improving energy monitoring at the consumer level without having to place sensors at device level, but rather use existing sensors at the whole building level. The concept of disaggregation is not new; however, only recently has it gained attention in the energy research domain. This is likely due to the emergence of smart meters and big data analytics.

Broadly speaking, disaggregation in essence is a single-channel source separation problem. The problem of recovering the components of an aggregate signal is an inverse problem and as such is, in general, ill-posed. Most disaggregation algorithms are batch algorithms and produce an estimate of the disaggregated signals given a batch of aggregate recordings. There have been a number of survey papers summarizing the existing methods (e.g. see [6], [7]). In an effort to be as self-contained as possible, we try to provide a broad overview of the existing methods and then explain how the disaggregation method presented in this paper differs from existing solutions.

The literature can be divided into two main approaches, namely, supervised and unsupervised. Supervised disaggregation methods require a disaggregated data set for training. This data set could be obtained by, for example, monitoring typical appliances using plug sensors. Supervised methods assume that the variations between signatures for the same type of appliances is less than that between signatures of different types of appliances. Hence, the disaggregated data set does not need to be from the building that the supervised algorithm is designed for. However, the disaggregated data set must be collected prior to deployment, and come from appliances of a similar type to those in the target building. Supervised methods are typically discriminative.

Unsupervised methods, on the other hand, do not require a disaggregated data set to be collected. They do, however, require hand tuning of parameters, which can make it hard for the methods to be generalized in practice. It should be said that also supervised methods have tuning parameters, but these can often be tuned using the training data.

The existing supervised methods include sparse coding [8], change detection and clustering based approaches [9], [10] and pattern recognition [11]. The sparse coding approach tries to reconstruct the aggregate signal by selecting as few signatures as possible from a library of typical signatures. Similarly, in our proposed framework we construct a library of dynamical models and reconstruct the aggregate signal by using as few as possible of these models.

The existing unsupervised methods include factorial hidden Markov models (HMMs), difference hidden Markov models and variants [12], [13], [14], [15], [16] and temporal motif mining [17]. Most unsupervised methods models the on/off sequences of appliances using some variation of HMMs. These methods do not make use of the signature of a device and assume that the power consumption is piecewise constant.

All method we are aware of lack the use of the dynamics of the devices. While the existing supervised methods often do use device signatures, these methods are discriminative and an ideal method would have a dynamical model that is capable to generating a device signature given a combination of initial state and input. Both HMMs and linear dynamical models are generative as opposed to discriminative, making them more advantageous for modeling complex system behavior. In the unsupervised domain, HMMs are used; however, they are not estimated using data and they do not model the signature of a device. The method we develop in this paper will combine the use of a generative model, i.e. linear dynamical models of devices, with a supervised approach to disaggregation.

III. DYNAMICAL MODELS

A. Framework

In our dynamical model framework, we model individual devices as single-input, single-output systems, where the output is the power consumed by the device and the input is the device usage. That is, the input is zero if a device is off, and the input is nonzero if the device is on. Thus, for device i, we have a model of the following form: $y_i = h_i(u_i)$, where y_i is the power consumption signal of the device, u_i is the input to the device, and h_i is a function that represents the underlying dynamics. We build a library of models which represent the appliance types we are interested in.

With a model for each device, we treat the total power consumption as the aggregate output of all devices, i.e. $y = \sum_{i=1}^{D} y_i$, where y is the total power consumption signal and D is the total number of device models. The task of disaggregation then translates into finding inputs for each device that generates our observed power consumption. In general, this solution will not be unique without more constraints on the input. Incorporating some prior on the form of the input, the problem becomes the following:

$$\begin{aligned} \arg\min_{\hat{y},u} \quad & L(\hat{y}, y_m) + g(u) \\ \text{subj. to} \quad & \hat{y}_i = h_i(u_i) \\ & \quad \text{for } i \in \{1, \ldots, D\} \\ & \hat{y} = \sum_{i=1}^{D} \hat{y}_i \end{aligned} \quad (1)$$

where y_m is the measured power consumption, $\hat{y}$ is the estimated power consumption, L is a loss function penalizing

deviations of $\hat{y}$ from y_m, and g is a regularization on the input that incorporates our priors.

B. Implementation

In this framework, the task of disaggregation can be broken down into two steps: system identification and disaggregation.

1) System identification: In the system identification step, we seek to build a library of models which represent all the devices we are interested in. We assume we are given time-series measurements of power consumption for individual devices, e.g. a toaster, a kettle, or a LCD projector, and we wish to find a model to capture the dynamics underlying the signal. This task has a deep history and well-established literature and results [18].

More specifically, for some device i, we are given T power usage samples, $y_i[k] \in \mathbb{R}$ for $k \in \{1, \ldots, T\}$, and a sequence of corresponding inputs, $u_i[k]$ for $k \in \{1, \ldots, T\}$. Assuming our world is causal, our goal is to find a satisfactory model such that $y_i[k] = h_{i,k}(u_i[1], \ldots, u_i[k])$.

Throughout this paper, we will use linear time-invariant (LTI) state-space models to represent the power consumption dynamics of individual devices, i.e. systems of the form:

$$
\begin{aligned}
x_i[k+1] &= A_i x_i[k] + b_i u_i[k] \\
y_i[k] &= c_i^{\mathrm{T}} x_i[k]
\end{aligned}
\tag{2}
$$

where n is the order of the device model and $x_i[k] \in \mathbb{R}^n$ for $k = 1, \ldots, T$ is a state underlying the dynamics. The framework generalizes to nonlinear, time-varying models as well, but for simplicity we merely consider the LTI case here.

Note that, under the assumption that similar devices have similar power consumption profiles, these models can be estimated offline. That is, for the task of disaggregation, we only need to estimate models for each class of devices once. Afterward, due to their generative nature, these models can be used for any household. Thus, this dynamical system framework would be cost-effective to deploy widely.

Furthermore, while power usage data can be easily recorded with plug sensors, it is not as convenient to record the input signal, $u_i[\cdot]$ for each plug. Thus, at this step, it may be necessary to apply blind system identification techniques, i.e. techniques for the case where both the system dynamics and the inputs are unknown. A detailed coverage of blind system identification is outside the scope of this paper; we refer the interested reader to the following references: [19], [20]. Also, the authors of this paper have also devised a method for blind system identification motivated by the disaggregation problem, see [21].

2) Disaggregation: With these dynamical models in hand, we can treat disaggregation as the task of finding an input that generates our observed output. The problem formulation is as follows. We are only given samples of aggregated power consumption for a household, $y_m[k] \in \mathbb{R}$ for $k = 1, \ldots, T$. Also, we know that the majority of the power consumption signal originates from a subset of our D modeled devices. We want to find inputs which result in a similar power consumption signal.

In this paper, we take the inputs of the system to be the device's setting when it is on. Take a conventional oven as an example. It can be off, or it could be on with a temperature setting that takes on continuous values. In this situation, the input is zero if the oven is off, or the input is the temperature setting if the oven is on. An important distinction is that the input is the temperature setting, *not* the temperature of the oven itself; the input can be thought of as a command to the device, e.g. if a user sets the oven to 350°F at time k^*, the input is $u_{\mathrm{oven}}[k] = 0$ for $k < k^*$ and $u_{\mathrm{oven}}[k] = 350$ for $k \geq k^*$. Looking at this example, we can see that a reasonable prior would be that the inputs u_i are piecewise constant, and that the changes in u_i across time are sparse. Throughout our paper, we use this as our prior on the inputs.

Returning to Equation 1, we define:

$$
\Delta u = \begin{bmatrix} u[1] - u[0] \\ u[2] - u[1] \\ \vdots \\ u[T] - u[T-1] \end{bmatrix}
\tag{3}
$$

and we take $g(x) = \mathrm{card}(x)$, i.e. the number of nonzero elements in x. Furthermore, we take L to be the Euclidean distance on $\mathbb{R}^T$. Thus, we have our optimization defined.

A common approach when one is trying to minimize the cardinality of a vector is to relax the cardinality into the ℓ_1 norm, which is convex. However, we found that this performs poorly in our framework. A likely explanation is that when a linear system is converted in the linear operator $\Delta u \mapsto y$, it will often fail to meet the desiderata for the ℓ_1 relaxation.

Another technique is necessary. First, we note that if we know which elements of Δu are nonzero, i.e. which devices turned on or off at what time, then it is easy to find the optimal Δu. We define each of these as a *configuration*. When $g(\cdot)$ is the cardinality operator, the optimal configuration is, informally, the configuration which results in the best fit with the fewest nonzero entries. However, finding this solution is combinatorial.

We seek relaxations which will make this optimization tractable. We assume that, at each time step, only one device turns on or off at a time. This is not an egregious assumption if our sampling rate is sufficiently large. Also, we assume that the devices switch on and off in sequence; a device does not turn on and then on again afterward. We can sort by time and place our possible configurations in a tree structure. More formally, at each time step, one of $D + 1$ things can happen: a device $d \in \{1, \ldots, D\}$ switches on or off, where only one of the two options is possible depending on its current configuration, or no device changes configuration. This induces a hierarchical ordering on configurations of different time intervals. That is, at depth T of the tree, the nodes are configurations at times $k \in \{1, \ldots, T\}$, and that node's children are configurations on $\{1, \ldots, T+1\}$.

If we think of the configuration at a given time as a mode, then this is a hybrid system estimation problem. The combinatorial problem above is often called a complete filter bank. This is still intractable, but we can use heuristics to

intelligently prune or merge the tree and keep the set of possible configurations manageable. For the general problem, pruning and merging methods are discussed in [22], [23], [24]. These methods are known as generalized pseudo-Bayesian filters or interacting multiple models. Also, note that these algorithms allow for disaggregation to be done online.

The disaggregation problem allows for several intuitive heuristics. First, if a given configuration continues to model the future data well, we assume no device changes state. Second, if the power consumption increases by a certain amount, a device is turning on. Finally, if the power consumption decreases by a significant amount, a device is turning off. These three heuristics are sufficient to make our optimization problem extremely efficient.

IV. SIMULATION

We implemented the disaggregation algorithm on simulated data. We generated $D = 5$ third-order single input, single output systems using MATLAB's `drss` function, normalized to have a DC gain of 1. The step responses for these 5 systems can be seen in Figure 1. Let the dynamics of each system be represented with matrices $A_i, b_i, c_i^{\mathrm{T}}, d_i$ for $i \in \{1, \ldots, D\}$. We assume we are given the true models for each of these D devices.
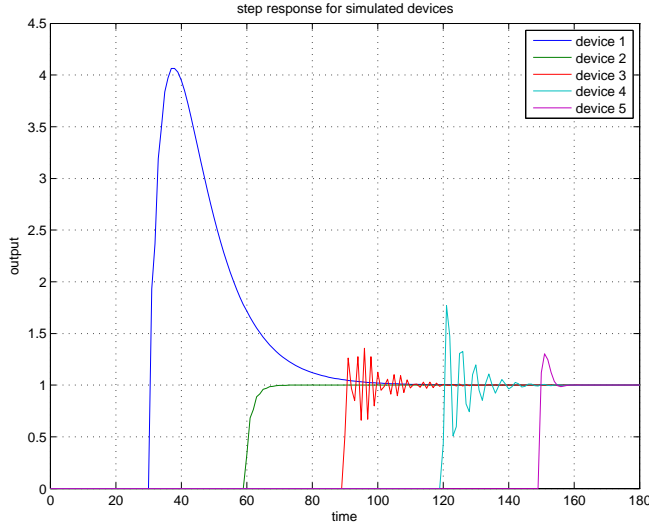

Fig. 1. The step responses of $D = 5$ randomly generated device models, spaced apart by 30 time steps.

We also observed that many real-life devices seem to have different dynamics when switching on and when switching off. For example, consider the root-mean-squre (RMS) current signal of a toaster, represented in Figure 2. There is overshoot when the toaster switches on, but the off dynamics do not show the same behavior. In fact, in all of the devices we measured, we found that when a device turns off, the power drops to a negligible amount almost instantly. We factor this into our simulated models as well.

Then, we created output signals for each system by using the inputs in Equation 4. These inputs were chosen to overlap
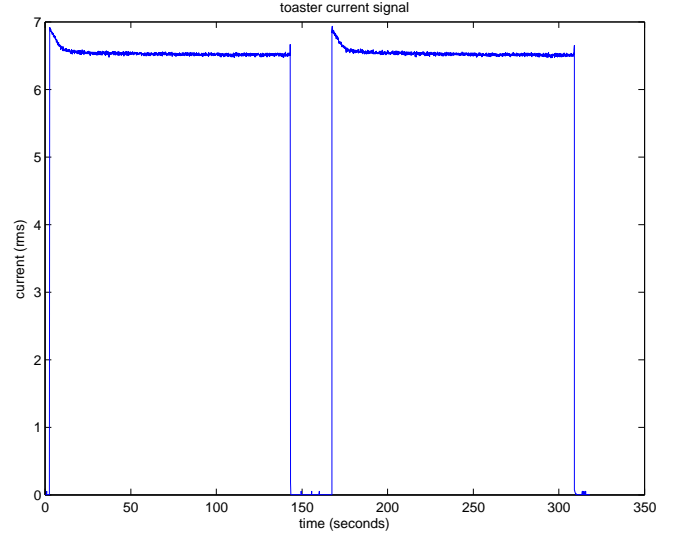

Fig. 2. The measured RMS current signal for a toaster. Note that the on-switches display overshoot while the off-switches do not.

significantly. Also, not every device is activated during the simulation. The aggregated signal was created by summing these individual inputs as well as white noise with 0 mean and 0.02 standard deviation.

$$
\begin{aligned}
u_1[k] &= 1.2 \quad \text{for } k \in \{20, \ldots, 100\} \\
u_2[k] &= 2 \quad \text{for } k \in \{130, \ldots, 400\} \\
u_3[k] &= 0.6 \quad \text{for } k \in \{180, \ldots, 300\} \\
u_4[k] &= 1.8 \quad \text{for } k \in \{250, \ldots, 350\} \\
u_i[k] &= 0 \quad \text{otherwise}
\end{aligned}
\tag{4}
$$

We then run the disaggregation method described in Section III-B. For simplicity, we assume that the input is zero initially. Then, as long as this configuration's expected output and the observed output are within a certain threshold, we keep the same configuration. When the observed output exceeds this threshold, we determine if the signal is increasing or decreasing. If it is increasing, we look at all devices and nearby times to find the device that best explains the change in the measured data, as well as nearby data afterward, when driven with a constant input. If it is decreasing, since all devices turn off in the same fashion, we determine which device turned off by looking at the contribution of each device in the estimated configuration.

More formally, let y_m be our measured signal, and let $\hat{y}$ be the predicted output under the estimated configuration. Suppose we detect a change at time k^* and let N be our lookahead time. Then, for nearby times k' and devices $i \in \{1, \ldots, D\}$ which are not currently on, we calculate:

$$
\begin{aligned}
\min_{y_i, x_i, u} \quad & \|e - y_i\|_2^2 \\
\text{subj. to} \quad & x_i[k+1] = A_i x_i[k] + b_i u \\
& \quad \text{for } k \in \{k', k'+2, \ldots, k^*+N-1\} \\
& x_i[k'] = 0 \\
& y_i[k] = c_i^T x_i[k] + d_i u \\
& \quad \text{for } k \in \{k', \ldots, k^*+N\}
\end{aligned}
\tag{5}
$$

where $e[k] = y_m[k] - \hat{y}[k]$ for $k \in \{k', \ldots, k^* + N\}$ is the deviation we need to explain. Note here that u is a scalar, not a time-dependent signal. That is, given k' and i, we find the best input magnitude to explain the behavior. Also, note that we are implicitly reducing the cardinality of Δu, as well as reducing the number of needed calculations, by only making these estimations when our estimated configuration is not satisfactory. Furthermore, if we wish to do online disaggregation, the lookahead parameter, N, determines how much delay is needed. The disaggregation estimate is:

$$
\begin{array}{rcll}
\hat{u}_1[k] & = & 1.2017 & \text{for } k \in \{20, \ldots, 100\} \\
\hat{u}_2[k] & = & 2.0104 & \text{for } k \in \{130, \ldots, 400\} \\
\hat{u}_3[k] & = & 0.5827 & \text{for } k \in \{180, \ldots, 300\} \\
\hat{u}_4[k] & = & 1.7987 & \text{for } k \in \{250, \ldots, 350\} \\
\hat{u}_i[k] & = & 0 & \text{otherwise}
\end{array}
\tag{6}
$$

Every device is successfully identified, and the switching times are also correctly identified. The simulated data is plotted against the estimated data in Figure 3.
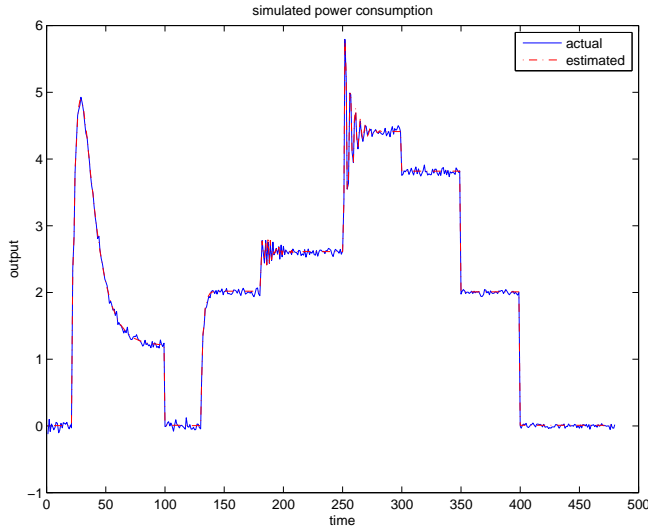

Fig. 3.   The simulated disaggregation results.

V. EXPERIMENT

For the verification of our disaggregation method, we deployed a small-scale experiment. To collect the data, we use the emonTx wireless open-source energy monitoring node from OpenEnergyMonitor[1]. We use current transformer (CT) sensors and an alternating current (AC) to AC power adapter to measure the current and voltage respectively of the devices that we monitored. For each device we measure the root-mean-square (RMS) current (I_{RMS}^i), RMS voltage (V_{RMS}^i), apparent power (P_{VA}^i), real power (P_{W}^i), power factor (θ_{pf}^i), and a UTC time stamp where the superscript i index denotes the ith device. The sampling rate is 12Hz.

For our experiment, we focused on small devices that would be featured in a residential or commercial office

[1] http://openenergymonitor.org/emon/emontx

building. First, we took individual plug-level measurements for a kettle, a toaster, a projector, a monitor, and a microwave. These devices consume anywhere from 70W to 1800W. We labeled the devices $\{1, \ldots, 5\}$, respectively. For the blind system identification of each of these devices, we used a simple change detection algorithm to generate input signals. Then, we fit autoregressive models with exogenous inputs.

Then, we ran an experiment where we had a microwave, a toaster, and a kettle (devices 5, 2, and 1, respectively) operating at different time intervals. These individual plug measurements are in Figure 4. We can note that the device power consumptions are not completely independent; one device turning on can affect the power consumption of another device. However, we found this effect to be negligible in our disaggregation algorithms.

Then, we ran an experiment where we had a microwave, a toaster, and a kettle (devices 5, 2, and 1, respectively) operating at different time intervals. These individual plug measurements are in Figure 4. We can note that the device power consumptions are not completely independent; one device turning on can affect the power consumption of another device. However, we found this effect to be negligible in our disaggregation algorithms.
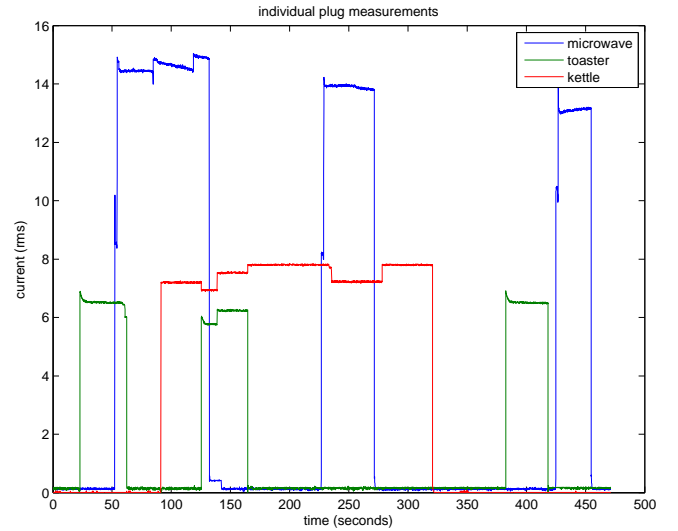

Fig. 4.   The measurements of individual plug RMS currents.

The results from using our disaggregation method on the experimental data is presented in Figure 5. The estimated power consumption lines up with the measured power consumption quite well. Furthermore, the power consumption of the toaster and the kettle are correctly identified. However, the microwave is erroneously identified as a monitor. This is because the dynamics of these two models are quite similar. This error can easily be compensated for by setting a maximum power consumption for each device. That is, we can state *a priori* that we know an LCD monitor will not draw over 10 amps of RMS current. When we add this prior, the microwave becomes correctly labeled.

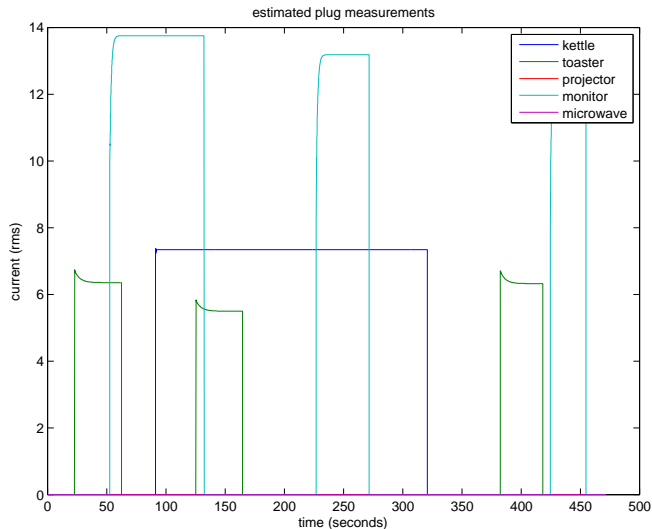Examining the data, we can see that methods which do

Fig. 5. The estimated power consumption signals of each device.

not take into account the dynamics of the devices, such as the hidden Markov models (HMM) methods in [13], [14], will likely confuse the kettle and toaster, which have similar amplitudes and can have similar durations. Also, the sparse coding method in [8] requires a large training data set to serve as a dictionary; here, we have a very small training set from which we derive system models. Thus, a direct comparison between our method and the sparse coding method is not possible.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present a novel framework to perform the task of disaggregation. We treat individual devices as systems and try to find the inputs which create the observed aggregated signal. This framework differs largely from the current disaggregation literature, which focuses largely on unsupervised methods. In contrast, our framework leverages many techniques and methods in system identification, optimal control, and hybrid system estimation.

We firmly believe that accounting for the power consumption profiles of individual devices will significantly improve disaggregation results. In a unsupervised setting, creating such models is very difficult. However, under the assumption that similar devices have similar power consumption profiles, the cost of collecting data and estimating these models is not significant. Thus, our framework, which utilizes more data than completely unsupervised methods, would not be infeasible to implement widely.

We tested an implementation of our framework on simulated data, as well as data from a small-scale experiment. The simulated data is completely recovered, and the experimental results closely matched the ground truth, although we did not achieve exact recovery. However, adding some reasonable assumptions allowed us to completely recover the ground truth.

For future work, we plan on deploying an experiment

where we collect measurements from more devices in an actual residential setting. In this experimental setting, we hope to learn not only device dynamics, but also the user's consumption patterns. One of the benefits of our framework is that we can learn devices independent of the consumer, and then learn the user's consumption patterns. Note that in many unsupervised methods, keeping the device constant while varying the consumer's usage patterns would result in different models entirely.

Additionally, throughout our experiments, we noticed that some devices do not fit our current modeling assumptions neatly. For example, the microwave warms up for a second or two, and begins heating. This results in two successive jumps in power consumption. With our current modeling assumptions, the best fit is an over-damped system. This is not ideal, and we hope to model devices as hybrid systems with multiple modes in the future.

VII. ACKNOWLEDGMENTS

REFERENCES

[1] L. Perez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[2] K. Ehrhardt-Martinez, K. A. Donnelly, and J. A. Laitner, "Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities," American Council for an Energy-Efficient Economy, Tech. Rep. Report Numer E105, June 2010.

[3] A. A. Cardenas, S. Amin, G. Schwartz, R. Dong, and S. Sastry, "A game theory model for electricity theft detection and privacy-aware control in ami systems," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012.

[4] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring," *Journal of Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[5] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *ASCI International Workshop on Computing in Civil Engineering*, 2009.

[6] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics,*, vol. 57, no. 1, pp. 76 –84, february 2011.

[7] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, 2011.

[8] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Neural Information Processing Systems*, 2010.

[9] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," *IEEE Computer Applications in Power,*, vol. 12, no. 4, pp. 47–51, 1999.

[10] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbé, and D. P. Seetharam, "Learning to be energy-wise: discriminative methods for load disaggregation," in *Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012.* IEEE, 2012, pp. 1–4.

[11] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[12] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *SDM'11*, 2011, pp. 747–758.

[13] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.

[14] M. J. Johnson and A. S. Willsky, "Bayesian Nonparametric Hidden Semi-Markov Models," Massachusetts Institute of Technology, Tech. Rep. arXiv:1203.1365, 2012.

[15] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Nonintrusive load monitoring using prior models of general appliance types," in *26th AAAI Conference on Artificial Intelligence*, 2012.

[16] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *11th International Conference on Machine Learning and Applications (ICMLA), 2012*, vol. 2. IEEE, 2012, pp. 515–520.

[17] H. Shao, M. Marwah, and N. Ramakrishnan, "A temporal motif mining approach to unsupervised energy disaggregation," in *Proceedings of 1st International Non-Intrusive Load Monitoring Workshop*, 2012.

[18] L. Ljung, *System Identification — Theory for the User*, 2nd ed. Upper Saddle River, N.J.: Prentice-Hall, 1999.

[19] K. Abed-Meraim, W. Qiu, and Y. Hua, "Blind system identification," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1310–1322, 1997.

[20] C. Li and S. Andersen, "Blind identification of non-gaussian autoregressive models for efficient analysis of speech signals," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, 2006, pp. I–I.

[21] H. Ohlsson, L. Ratliff, R. Dong, and S. S. Shankar, "Blind identification of arx models with piecewise constant inputs," in *Proceedings of the 52th IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013, submitted to.

[22] I. Hwang, H. Balakrishnan, and C. Tomlin, "State estimation for hybrid systems: applications to aircraft tracking," *Control Theory and Applications, IEE Proceedings -*, vol. 153, no. 5, pp. 556–566, Sept.

[23] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *Automatic Control, IEEE Transactions on*, vol. 33, no. 8, pp. 780–783, Aug.

[24] Y. Bar-Shalom, K. Chang, and H. Blom, "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 296–300, Mar.